AMENDMENTS TO THE CLAIMS:

1. (Currently amended) A method of executing a linear algebra subroutine on a machine having at least one floating point unit (FPU) with one or more associated load/store units (LSU) to load data into and out of floating point registers (FRegs) of said FPU, said method comprising:

for an execution code controlling an operation of said floating point unit (FPU) performing a linear algebra subroutine execution involving three matrix operands in a level 3 nested loop matrix-matrix type kernel type operation (level 3 processing), moving data in a contiguous and stride one ~~format~~ manner into a cache providing data for said FPU for direct loading into said FPU, so that said LSUs ~~can~~ load said data into said FRegs before it is to be used in said linear algebra subroutine execution, said data being prefetched into said cache from a memory in a register block format, said register block format predetermined to provide a single data stream for each of said three matrix operands involved in said level 3 processing, thereby providing only three streams of data for said level 3 processing, each said stream comprising contiguous data, and to allow a loading of these three streams into said ~~FPU~~ FRegs by said LSUs,

said register block format comprising a data storage format wherein data is stored in blocks of size p-by-q, where p and q are small integers, meaning that p and q are sufficiently small so that pieces of all data words of a block of size p-by-q of one or more of these blocks can be fitted into said FRegs as a result of one or more single instructions, ~~said~~ each block comprising contiguous data to be moved stride one, ~~and~~ said format of said register blocks being predetermined to preclude a data copy processing during said level 3 processing,

wherein a processor of said machine preliminarily selects, from a plurality of six kernels, two kernels to use for executing said level 3 processing, said six kernels respectively

having different prefetch patterns, said six kernels adapted to perform said level 3 processing

using said register block format, and whereby said two kernels are selected so that said data

copy processing during said level 3 processing is not necessary, and

wherein said machine comprises M (M ≥ 1) levels of caches and a main memory and

said three data streams ~~comprise one~~ , as comprising said single stream of data ~~of one~~ for

each matrix operand of said level 3 ~~processing, as~~ processing, are considered to be resident in

said ~~cache~~ caches and main memory ~~and one stream each for data for two remaining matrix~~

~~operands of said level 3 processing, as considered to respectively be residing in a memory or~~

~~a cache having a level higher than or equal to a level of said cache~~ as dictated by a logic of

said two kernels.

2. (Previously presented) The method of claim 1, wherein said moving data is accomplished

by scheduling move type instructions into time slots existing in a Level 3 Dense Linear Algebra

Subroutine.

3. (Previously presented) The method of claim 1, wherein said linear algebra subroutine

comprises a matrix multiplication operation.

4. (Previously presented) The method of claim 1, wherein said linear algebra subroutine

comprises an equivalent of a subroutine from LAPACK (Linear Algebra PACKage).

5. (Previously presented) The method of claim 1, wherein said linear algebra subroutine

invokes a BLAS Level 3 L1 cache kernel.

6. (Currently amended) An apparatus, comprising:

a memory system to store matrix data to be used for processing in a linear algebra program;

a floating point unit (FPU) to perform said processing; and

a load/store unit (LSU) to load data to be processed by said FPU, said LSU loading said data into a plurality of floating point registers (FRegs) of said FPU; and .

wherein said memory system comprises a main memory and M (M ≥ 1) levels of cache, including a cache to store that data from said memory and provide provides said data directly into said FRegs using said LSU,

wherein said matrix data in said main memory is moved by into said cache prior to a need for said data to be loaded by said LSU into said FRegs for said processing, said data being prefetched into said cache from said main memory in a format predetermined to provide only three data streams for a level 3 nested loop matrix-matrix type kernel type operation (level 3 linear algebra processing), and to allow a stride one loading of these streams into said FPU FRegs by said LSU using SIMD (single instruction, multiple data) k ≥ 1, k being a number of data elements involved in said single instruction instructions, and to preclude a data copy processing during said level 3 linear algebra processing,

wherein said format comprises a register block format wherein data is stored in blocks of size p-by-q, where p and q are small integers, meaning that pieces of all data words of a block of size p-by-q of one or more of these blocks can be fitted into said FRegs as a result of one or more single instructions,

wherein a processor of said apparatus preliminarily selects, from a plurality of six kernels, two kernels to use for executing said level 3 linear algebra processing, said six kernels respectively having different prefetch patterns, said six kernels adapted to perform

said level 3 linear algebra processing using said register block format, thereby said two kernels are selected so that said data copy processing during said level 3 linear algebra processing is not necessary, and

wherein said three data streams comprise ~~only~~ one stream each of data ~~of one~~ for each matrix of three matrices involved in said level 3 linear algebra processing, as considered to be resident in said ~~cache, and one stream each for data for two remaining matrix operands of said level 3 linear algebra processing, as considered to respectively reside in a memory or a cache level higher than or equal to a level of said cache~~ caches and main memory as dictated by a logic of said two kernels.

7. (Original) The apparatus of claim 6, wherein said linear algebra program comprises a matrix multiplication operation.

8. (Previously presented) The apparatus of claim 6, wherein said linear algebra program comprises an equivalent of a subroutine from LAPACK (Linear Algebra PACKage).

9. (Previously presented) The apparatus of claim 6, wherein said processing comprises invoking a BLAS Level 3 L1 cache kernel.

10. (Canceled)

11. (Previously presented) The apparatus of claim 6, wherein said moving instructions are inserted into time slots existing in a Level 3 Dense Linear Algebra Subroutine.

12. (Currently amended) A computer-readable storage medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method of executing linear algebra subroutines on a SIMD (single instruction, multiple data) machine having at least one floating point unit (FPU) with one or more associated load/store units (LSUs) to load data into and out of floating point registers (FRegs) of said at least one FPU by way of a cache, said method comprising:

for an execution code controlling an operation of a floating point unit (FPU) performing a linear algebra subroutine execution, moving data into said cache providing said data into said FPU in a contiguous and stride one manner into said cache for direct loading into said FPU,

wherein said data is prefetched into said cache from a main memory in a format predetermined to provide only three data streams for a level 3 nested loop matrix-matrix type kernel type operation (level 3 linear algebra processing), using a stride one loading of these three streams into said FPU by said LSUs with SIMD (single instruction, multiple data) k→ 1, k being a number of data elements involved in said single instruction instructions, and to provide data so that a data copy processing during said level 3 linear algebra processing is not necessary,

wherein said format comprises a register block format wherein data is stored in blocks of size p-by-q, where p and q are small integers, meaning that pieces of all data words of a block of size p-by-q of one or more of these blocks can be fitted into said FRegs as a result of one or more single instructions, each said block comprising contiguous data to be moved stride one,

wherein a processor of said digital processing apparatus preliminarily selects, from a plurality of six kernels, two kernels to use for executing said level 3 linear algebra

9

processing, said six kernels respectively having different prefetch patterns, said six kernels adapted to perform said level 3 linear algebra processing using said register block format,

    wherein said digital processing apparatus comprises M (M ≥ 1) levels of caches and said main memory, said processor further preliminarily selecting, from said plurality of six kernels, two kernels to use for executing said level 3 matrix multiplication processing as data streams from different levels of said M levels of cache, said two kernels being selected so that a data copy processing during said level 3 processing is not needed and

    wherein said three data streams comprise ~~one~~ a single stream of data for each ~~of one~~ matrix of three matrices involved in said level 3 linear algebra processing, as considered to be resident in said ~~cache, and one stream each for data for two remaining streams, meaning one stream each for remaining two matrix operands of said level 3 linear algebra processing, as considered to respectively reside in a memory or a cache having a level higher than or equal to a level of said cache~~ caches and main memory as dictated by a logic of said two kernels.

13. (Previously presented) The computer-readable storage medium of claim 12, wherein said moving data is accomplished by inserting move type instructions into time slots existing in a Level 3 Dense Linear Algebra Subroutine.

14. (Previously presented) The computer-readable storage medium of claim 12, wherein said linear algebra subroutine comprises a matrix multiplication operation.

15. (Previously presented) The computer-readable storage medium of claim 12, wherein said linear algebra subroutine comprises an equivalent of a subroutine from LAPACK (Linear Algebra PACKage).

16. (Previously presented) The computer-readable storage medium of claim 12, wherein said linear algebra subroutine invokes a BLAS Level 3 L1 cache kernel.

17-20. (Canceled)